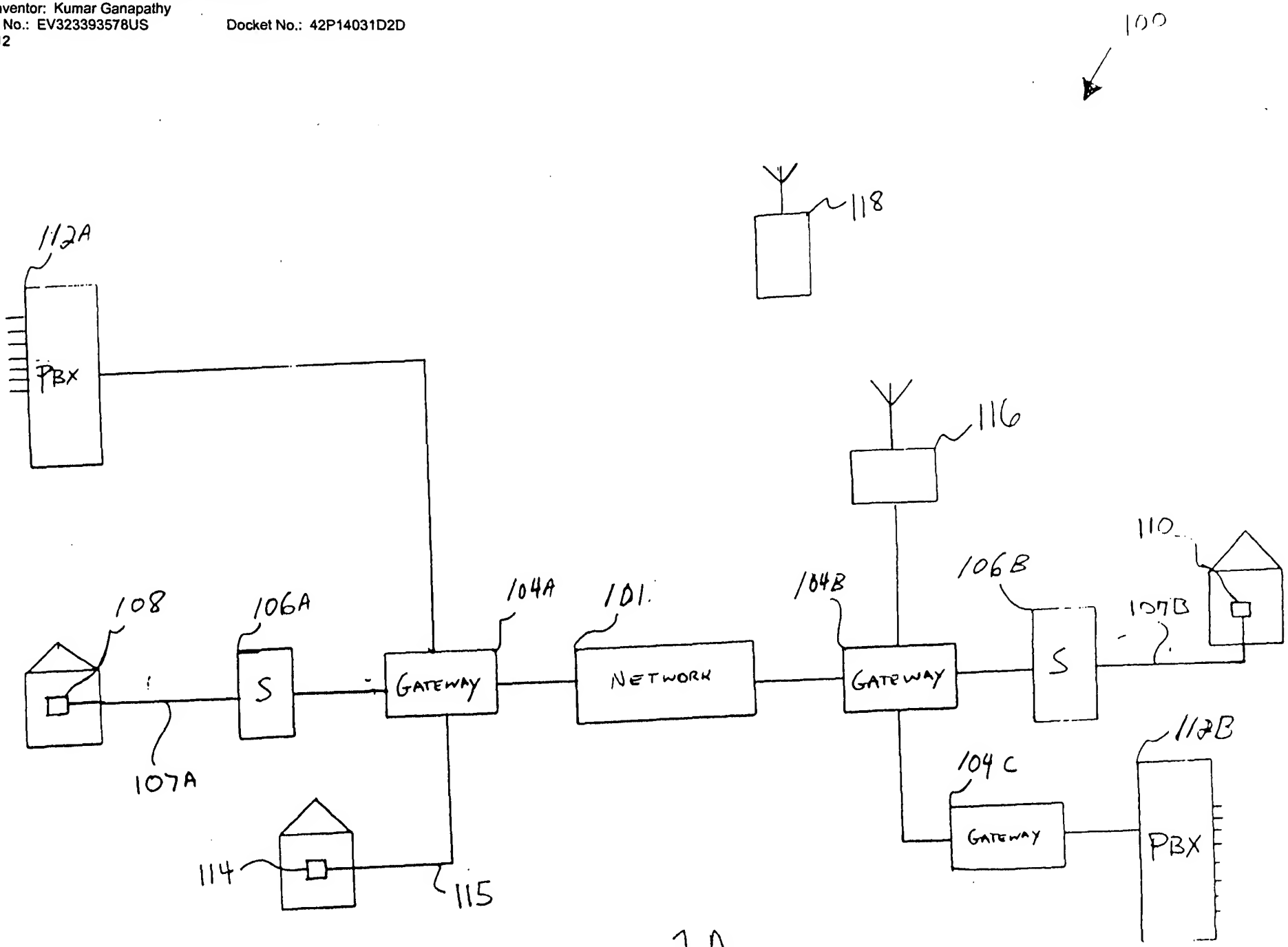
100
118
112A
PBX
116
110
106B
108
106A
104A
101
104B
107B
S
GATEWAY
NETWORK
GATEWAY
S
107A
112B
104C
GATEWAY
PBX
114
115

FIG. 1A

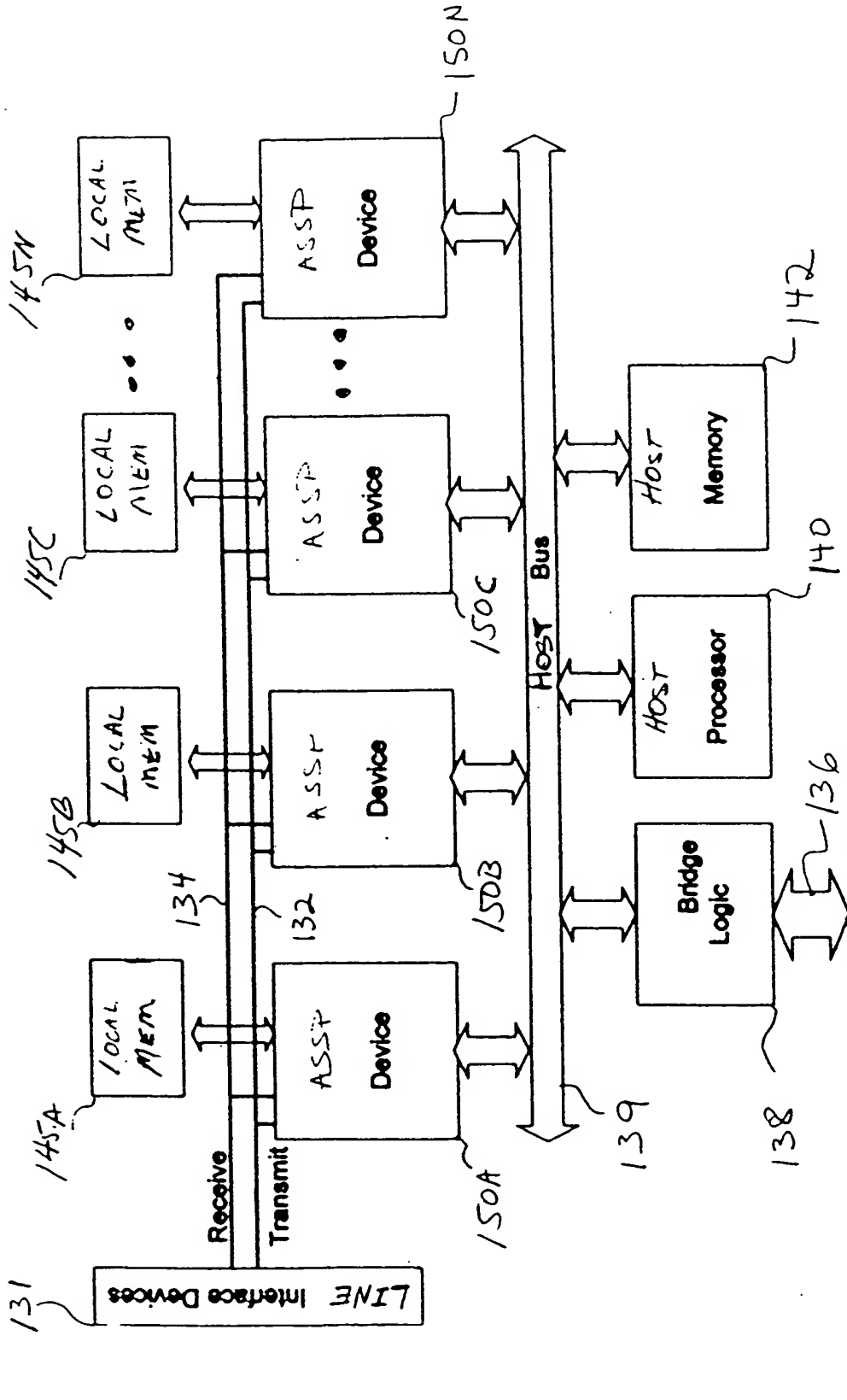
130
131
LINE Interface Devices
Receive
Transmit
134
132
136
138
139
140
142
Host Bus
Host Processor
Host Memory
Bridge Logic
BACKPLANE
NETWORK INTERFACE BUS
ASSP Device
LOCAL MEM
145A
145B
145C
145N
150A
150B
150C
150N

FIG. 1B

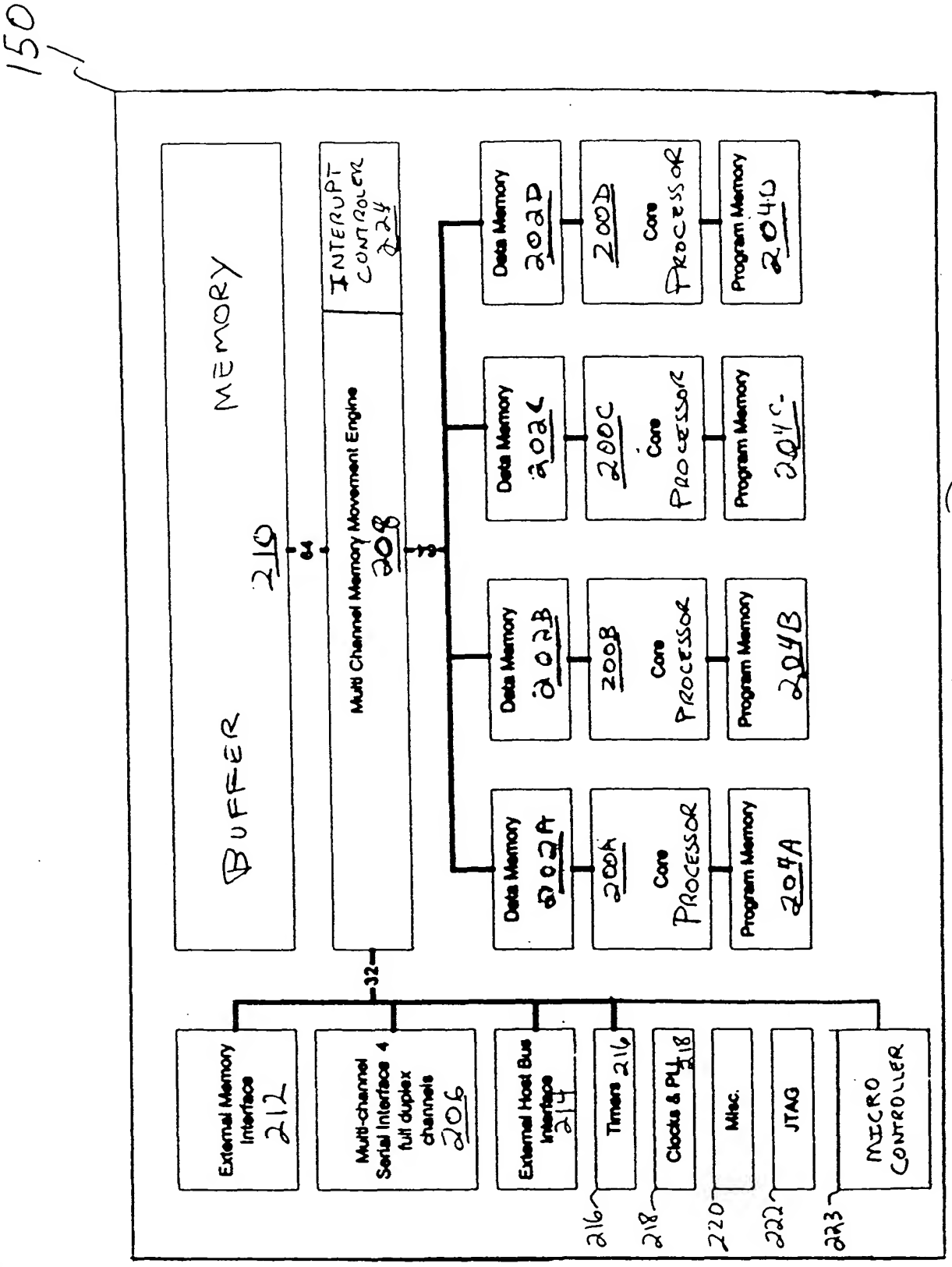
150
MEMORY
BUFFER
210
64
INTERUPT CONTROLLER 224
Multi Channel Memory Movement Engine 208
3
Data Memory 202D
200D Core Processor
Program Memory 204D
Data Memory 202C
200C Core Processor
Program Memory 204C
Data Memory 202B
200B Core Processor
Program Memory 204B
Data Memory 202A
200A Core Processor
Program Memory 204A
32
External Memory Interface 212
Multi-channel Serial Interface 4 full duplex channels 206
External Host Bus Interface 214
Timers 216
Clocks & PLL 218
Misc.
JTAG
MICRO CONTROLLER
216
218
220
222
223

FIG. 2

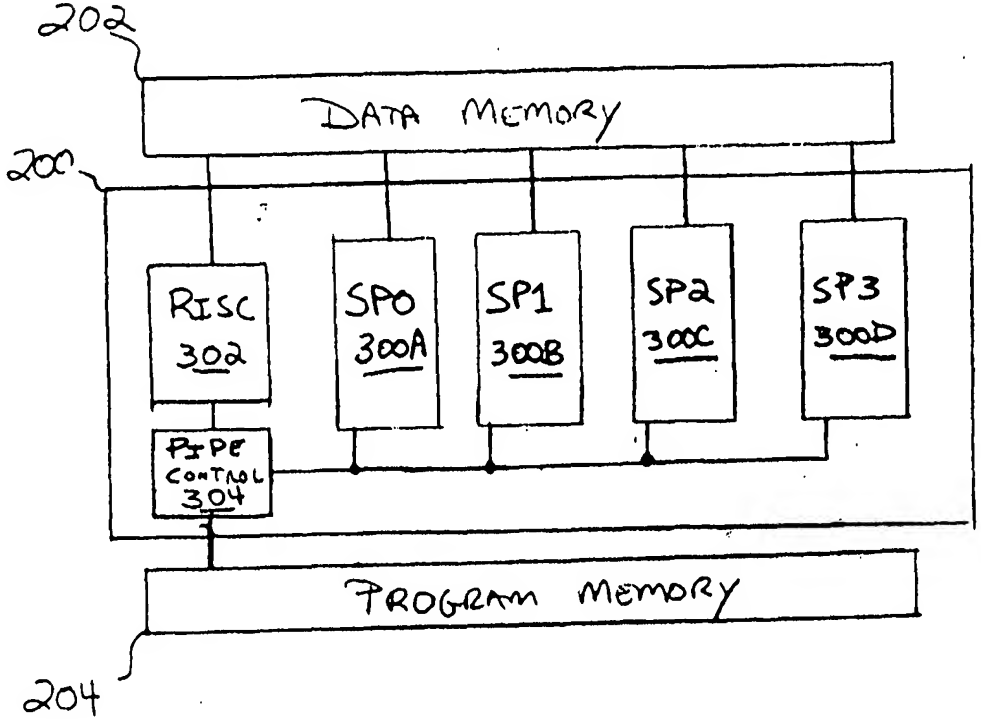

FIG. 3

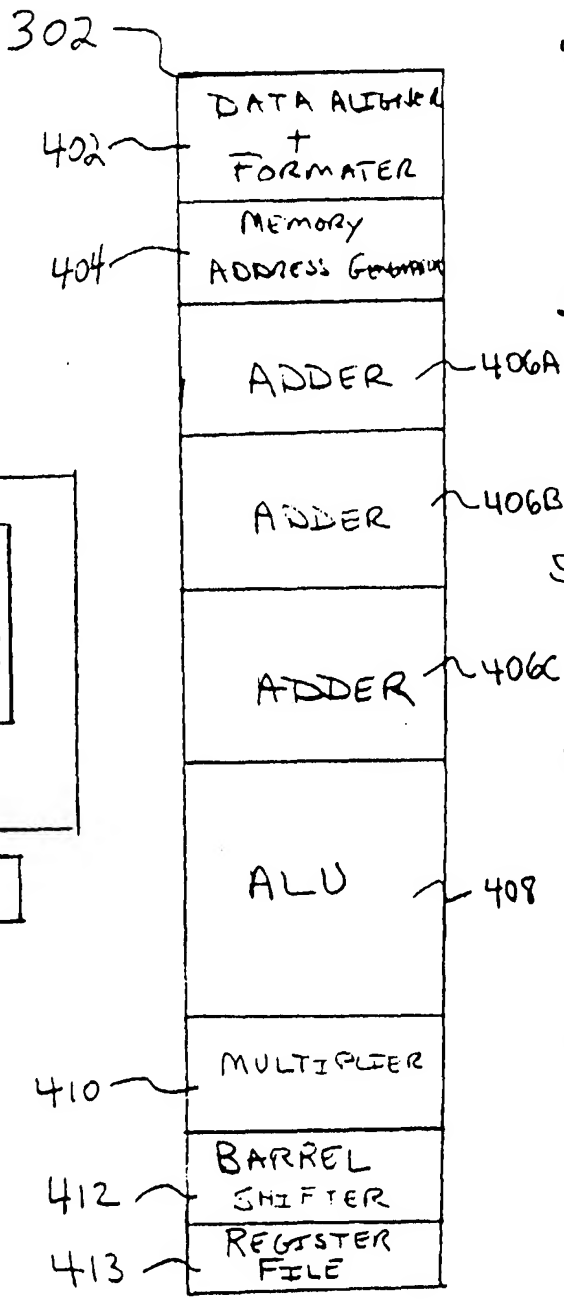

FIG. 4

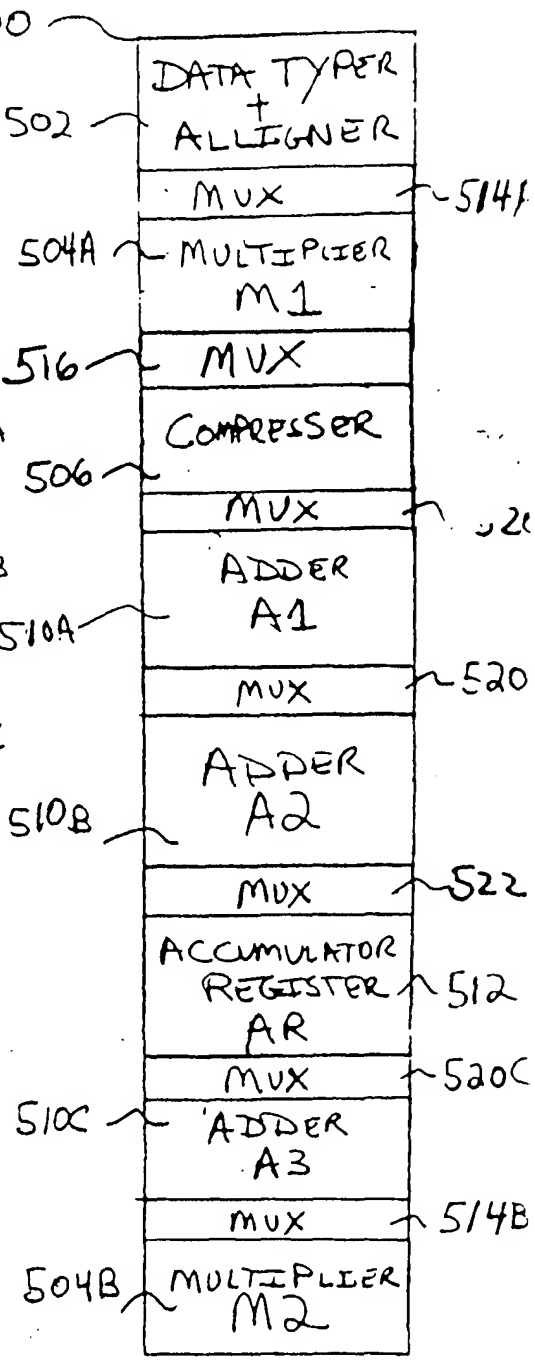

FIG. 5A

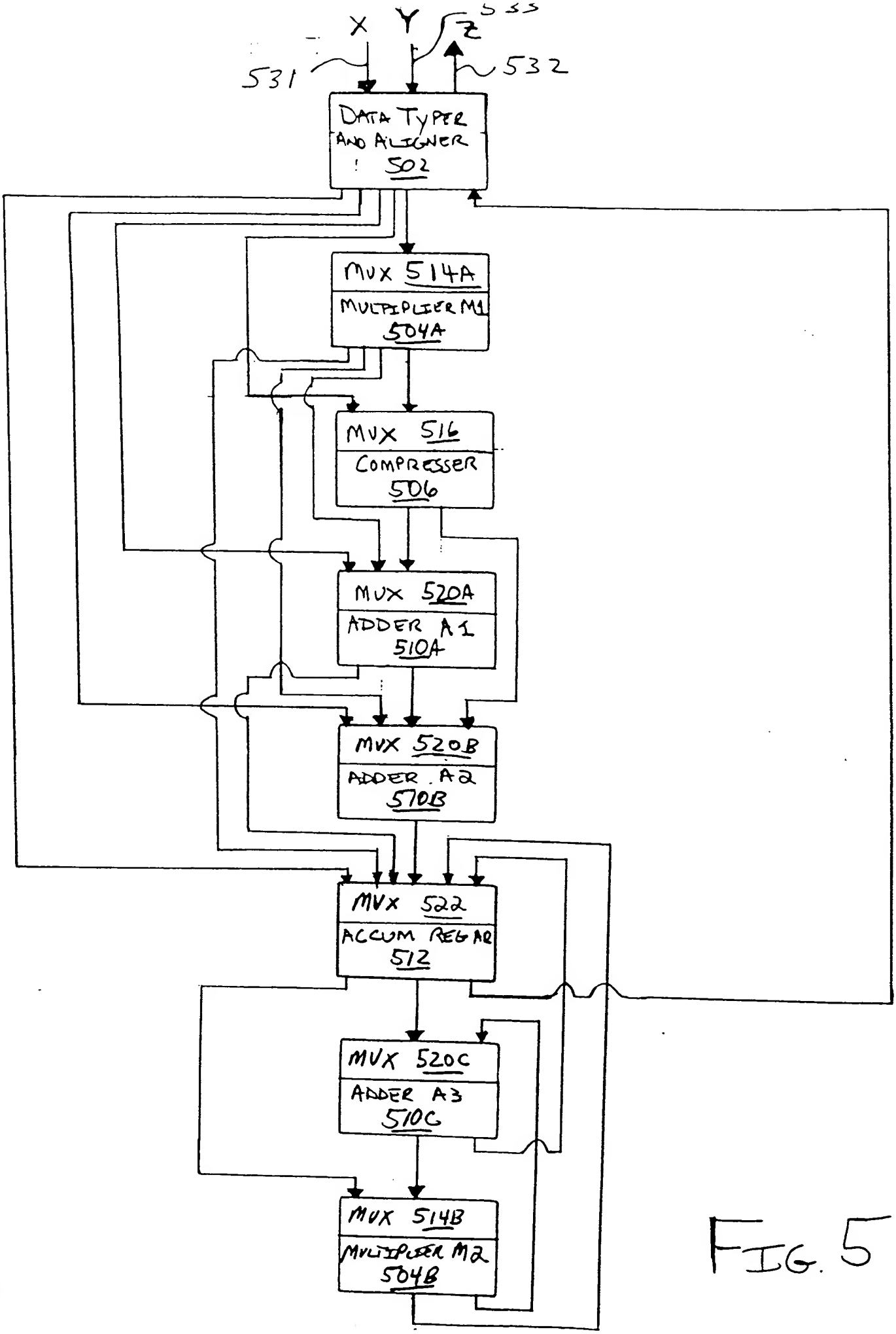
X
Y
Z
533
531
532
DATA TYPER AND ALIGNER 502
MUX 514A
MULTIPLIER M1 504A
MUX 516
COMPRESSER 506
MUX 520A
ADDER A1 510A
MUX 520B
ADDER A2 510B
MUX 522
ACCUM REG AR 512
MUX 520C
ADDER A3 510C
MUX 514B
MULTIPLIER M2 504B

FIG. 5B

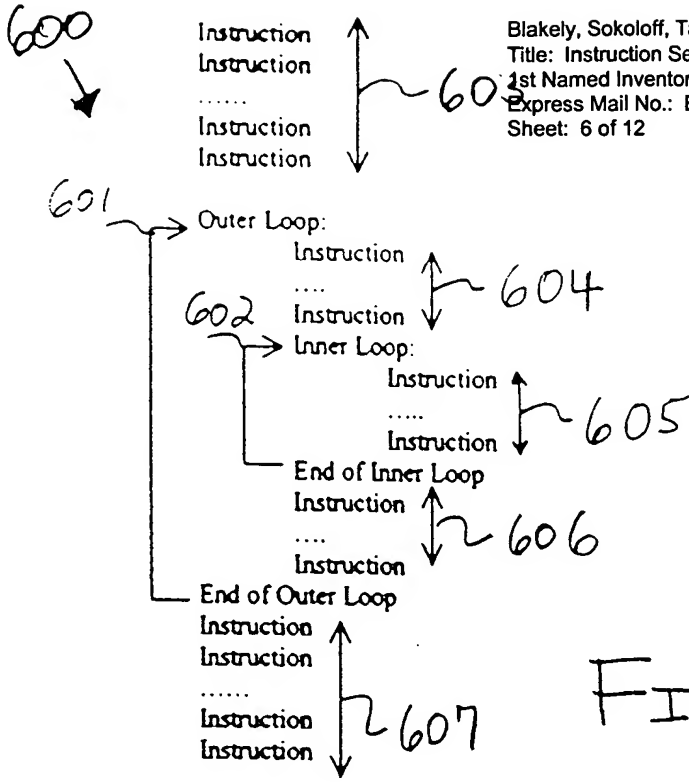

FIG. 6A

610

611 MAIN OP	612 SUB OP
MULT	NOP
ADD	MIN/MAX
MIN/MAX	ADD
NOP	MULT

FIG. 6B

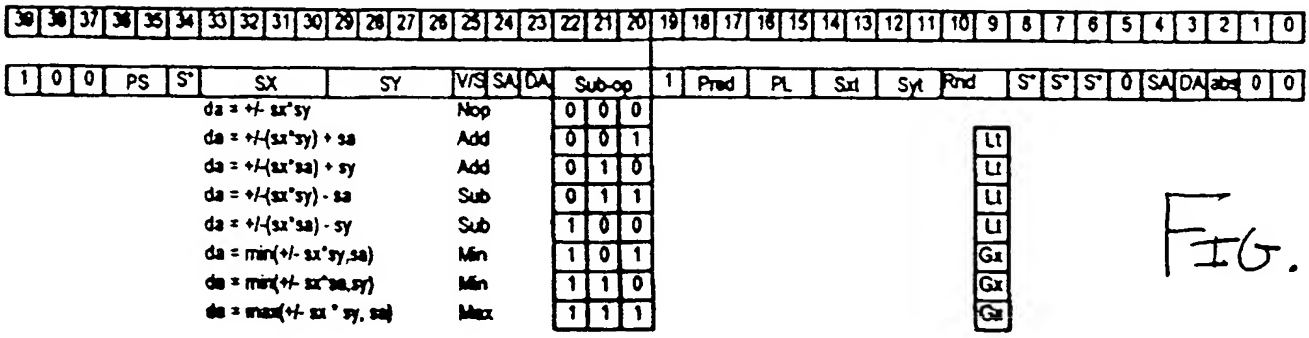

FIG. 6C

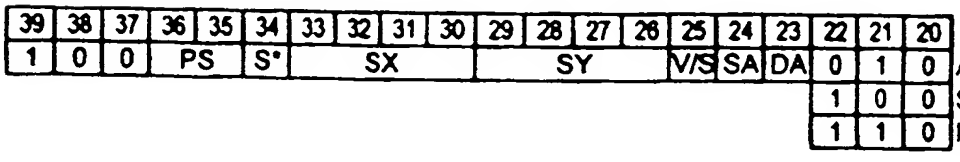

FIG. 6D

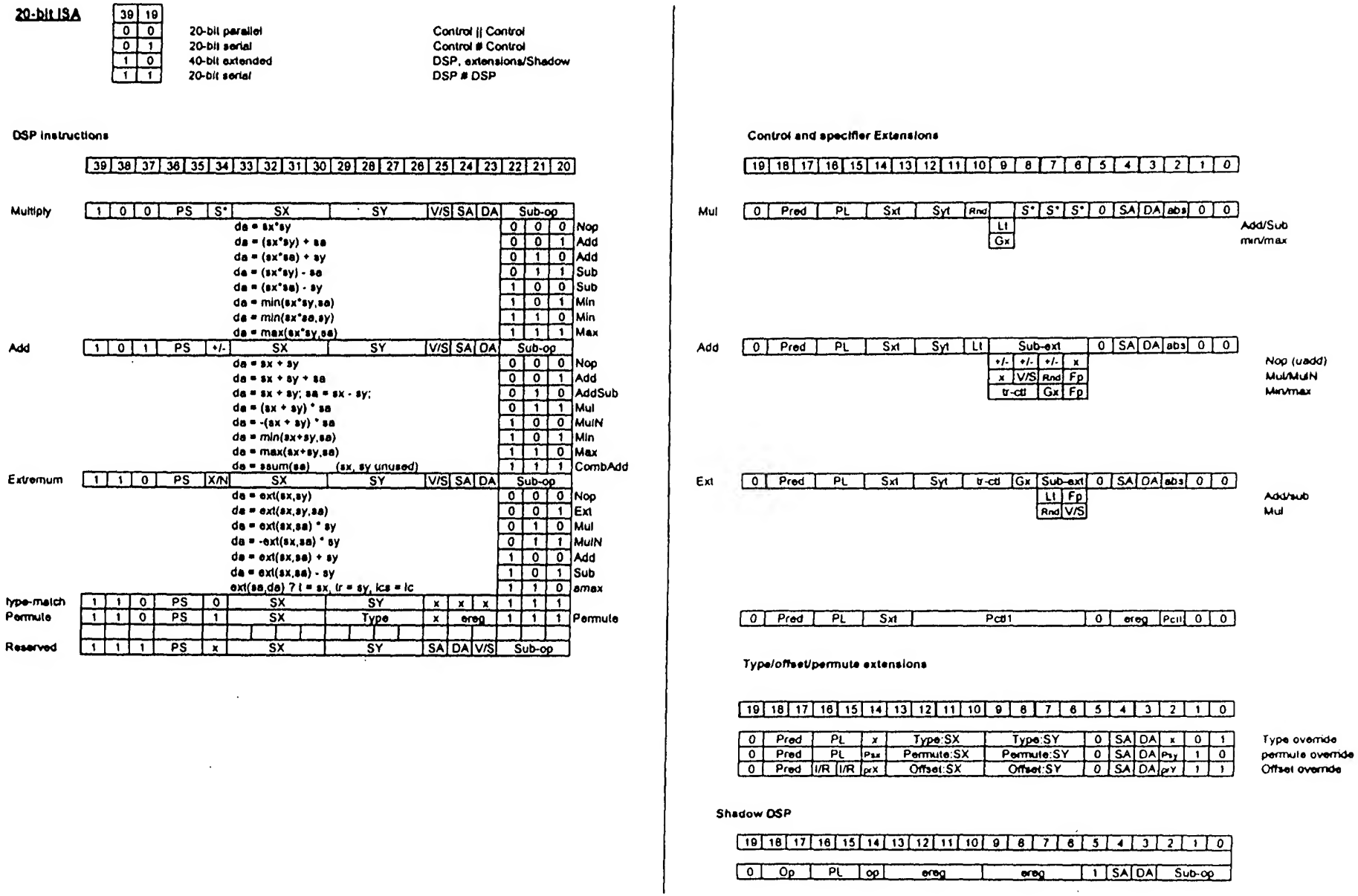
20-bit ISA
39 19
0 0 20-bit parallel Control || Control
0 1 20-bit serial Control # Control
1 0 40-bit extended DSP, extensions/Shadow
1 1 20-bit serial DSP # DSP
DSP Instructions
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20
Multiply 1 0 0 PS S* SX SY V/S SA DA Sub-op
da = sx*sy 0 0 0 Nop
da = (sx*sy) + se 0 0 1 Add
da = (sx*se) + sy 0 1 0 Add
da = (sx*sy) - se 0 1 1 Sub
da = (sx*se) - sy 1 0 0 Sub
da = min(sx*sy,se) 1 0 1 Min
da = min(sx*se,sy) 1 1 0 Min
da = max(sx*sy,se) 1 1 1 Max
Add 1 0 1 PS +/- SX SY V/S SA OA Sub-op
da = sx + sy 0 0 0 Nop
da = sx + sy + se 0 0 1 Add
da = sx + sy; se = sx - sy; 0 1 0 AddSub
da = (sx + sy) * se 0 1 1 Mul
da = -(sx + sy) * se 1 0 0 MulN
da = min(sx+sy,se) 1 0 1 Min
da = max(sx+sy,se) 1 1 0 Max
da = saum(se) (sx, sy unused) 1 1 1 CombAdd
Extremum 1 1 0 PS X/N SX SY V/S SA DA Sub-op
da = ext(sx,sy) 0 0 0 Nop
da = ext(sx,sy,se) 0 0 1 Ext
da = ext(sx,se) * sy 0 1 0 Mul
da = -ext(sx,se) * sy 0 1 1 MulN
da = ext(sx,se) + sy 1 0 0 Add
da = ext(sx,se) - sy 1 0 1 Sub
ext(se,de) ? t = sx, tr = sy, ics = ic 1 1 0 amax
type-match 1 1 0 PS 0 SX SY x x x 1 1 1
Permute 1 1 0 PS 1 SX Type x ereg 1 1 1 Permute
Reserved 1 1 1 PS x SX SY SA DA V/S Sub-op
Control and specifier Extensions
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Mul 0 Pred PL Sxt Syt Rnd Lt Gx S* S* S* 0 SA DA abs 0 0
Add/Sub min/max
Add 0 Pred PL Sxt Syt Lt Sub-ext 0 SA DA abs 0 0
+/- +/- +/- x
x V/S Rnd Fp
tr-ctl Gx Fp
Nop (uadd) Mul/MulN Min/max
Ext 0 Pred PL Sxt Syt tr-ctl Gx Sub-ext 0 SA DA abs 0 0
Lt Fp
Rnd V/S
Add/sub Mul
0 Pred PL Sxt Pctl 0 ereg Pctl 0 0
Type/offset/permute extensions
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 Pred PL x Type:SX Type:SY 0 SA DA x 0 1 Type override
0 Pred PL Psx Permute:SX Permute:SY 0 SA DA Psy 1 0 permute override
0 Pred I/R I/R pX Offset:SX Offset:SY 0 SA DA pY 1 1 Offset override
Shadow DSP
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 Op PL op ereg ereg 1 SA DA Sub-op

FIG. 6 E

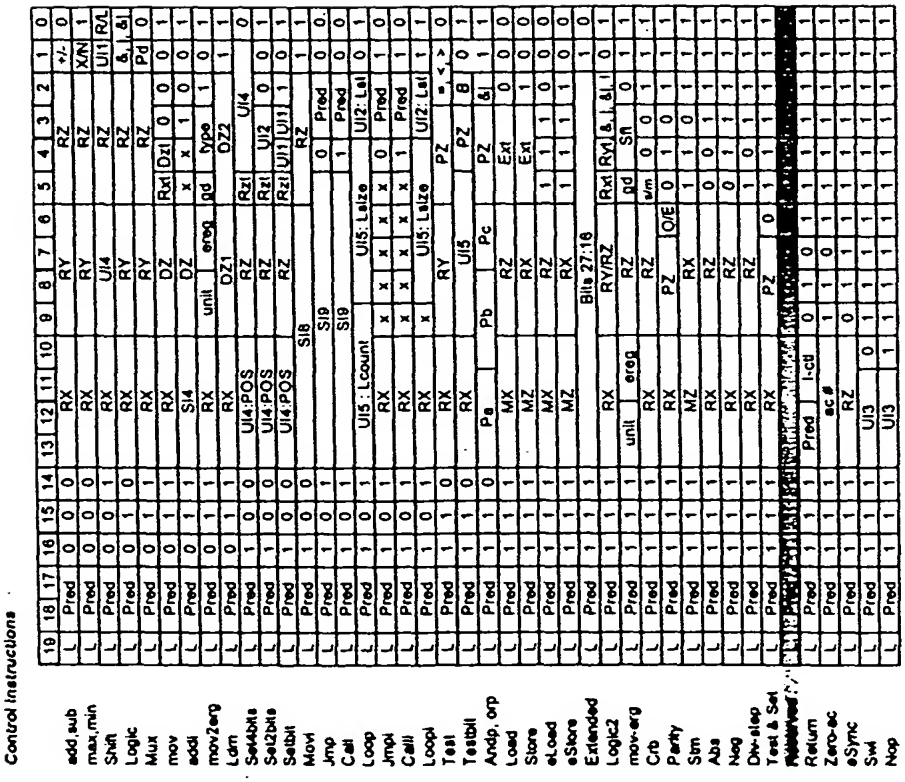
Control Instructions
<Bit1, Bits9-6> == UI5 (Shift Amount)
<Bit3, Bits13-10> == UI5 :POS

FIG. 6F

Extended Control

	Bits 13:2 of upper half (39:20)																												
	13	12	11	10	9	8	7	6	5	4	3	2	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Insert/Extract	RX				RZ				0	0	0	0	0	x	x	0	Rxt	Rzt	I/E	R/I	R/I	Offset:UI5					Length:UI5					0
																						x	RY				RV				x	
Insert	UI4: length				RZ				0	0	0	1	0	x	x	0	Rzt	UI5: Position				Imm10										
Shift	RX				RZ				0	0	0	0	0	nzh	nzl	0	rzh	rzl	D	U/S	1	Shift: UI5					A/L	L/I	R/L	0	Fill	1
																					0	ryh	RY									
Rotate	RX				RZ				0	0	0	0	0	x	x	0	x	x	x	x	1	Shift: UI5					1	x	R/L	1	x	1
																					0	ryh	RY									
jmp, call	UI7							J/C	0	0	1	0	0	Pred		0	x	UI15														
dloop	UI4: outer LC				UI4: inner LC				0	0	1	1	0	x	exit	0	UI1	UI4: outer Lsize				UI4: inner Lsize				UI2: O-Ls	UI4: inner Lstart				0
dloopi	RX				RY				0	0	1	1	0	x	exit	0	x	UI4: outer Lsize				UI4: inner Lsize				UI2: O-Ls	UI4: inner Lstart				1
mult	RX				RY				0	1	0	0	0	x	x	0	0	nzh	md	ryh	+/-	+/+	RZ				L/I	rzh	rzl	s/u	s/u	0
add/sub	RX				RY				0	1	0	0	0	x	x	0	0	nzh	nd	ryh	ryl	+/-	RZ				L/I	rzh	rzl	x	x	1
Reserved									0	1	0	0	0	x	x	0	0															1
logicp	PX			D	PZ				0	1	0	0	0	x	x	0	1	T/F	T/F	T/F	&I	&I	PY				PV				x	1
Testi	RX			D	PZ				0	1	0	1	0	=,>,<		0	Imm16															
Movi	H/L	Fill			RZ				0	1	1	0	0	x	x	0	Imm16															
loadi	Type				RZ				0	1	1	1	0	x	x	0	0	0	Imm14														
storei	Type				RX				0	1	1	1	0	x	x	0	0	1	Imm14														
loadi	RX				RZ				0	1	1	1	0	x	x	0	1	Rzt	0	Type		SI10										
storei	MZ				RX				0	1	1	1	0	x	x	0	1	Rzt	1	Type		SI10										
Addi/subi	RX				RZ				1	0	+/-	0	0	L/I	s/u	0	Imm16															
mini,maxi	RX				RZ				1	0	X/N	1	0	x	x	0	Imm16															
andi,ori	RX				RZ				1	1	&I	H/L	0	x	x	0	Imm16															

Fill: Sign/Zero

Bit 15 is continuation of inner LC

andp, orp, andorp, orandp: pz = (px relop py) relop pv)

FIG. 6G

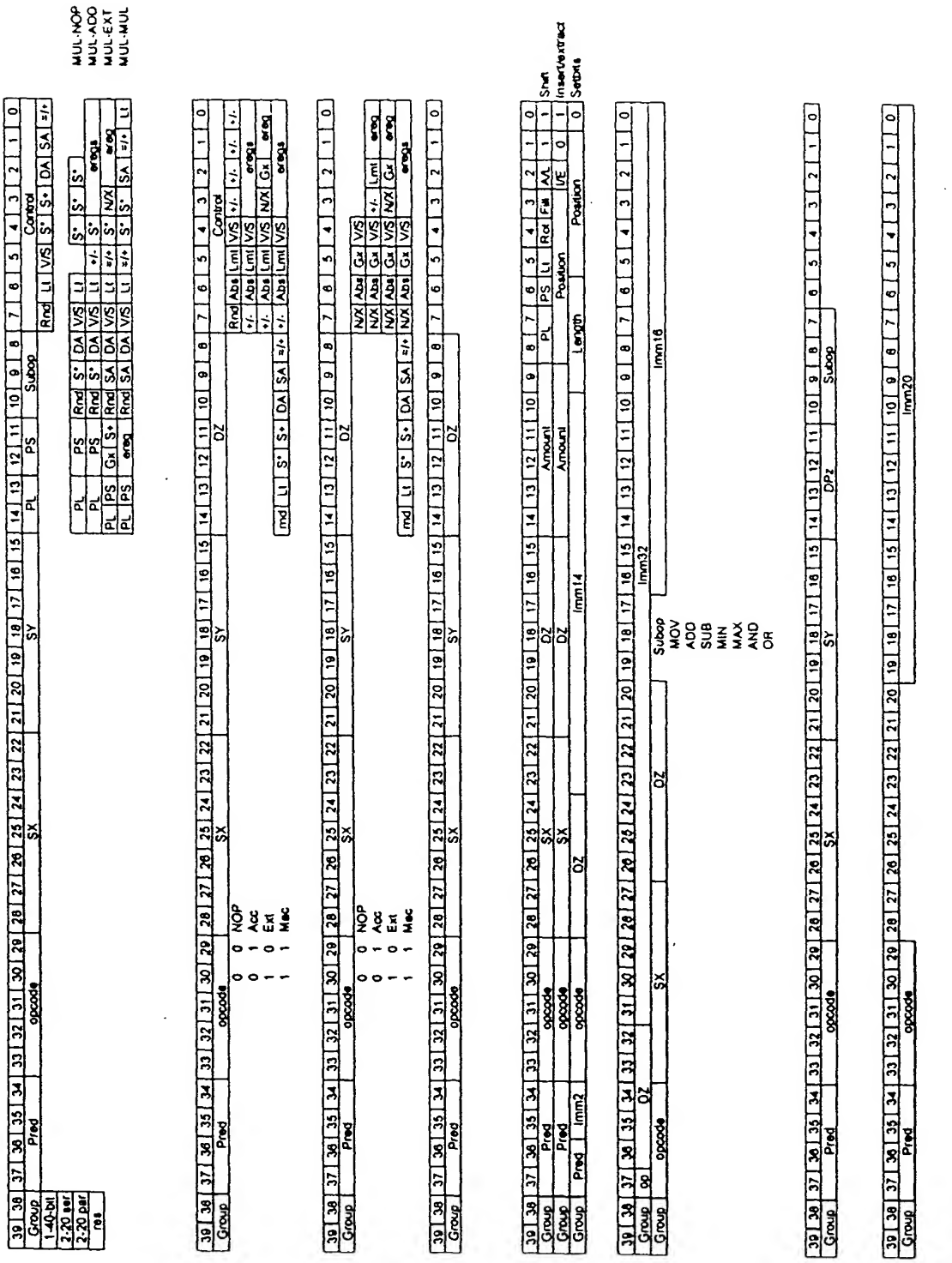
MAC:
ARITH:
EXT:
LOGIC:
SHIFT:
Immediate:
Test:
Branch:
Misc:
MUL-NOP
MUL-ADD
MUL-EXT
MUL-MUL
NOP
Acc
Ext
Mac
MOV
ADD
SUB
MIN
MAX
AND
OR
Shift
Insert/extract
Set/Clr

FIG. 6H

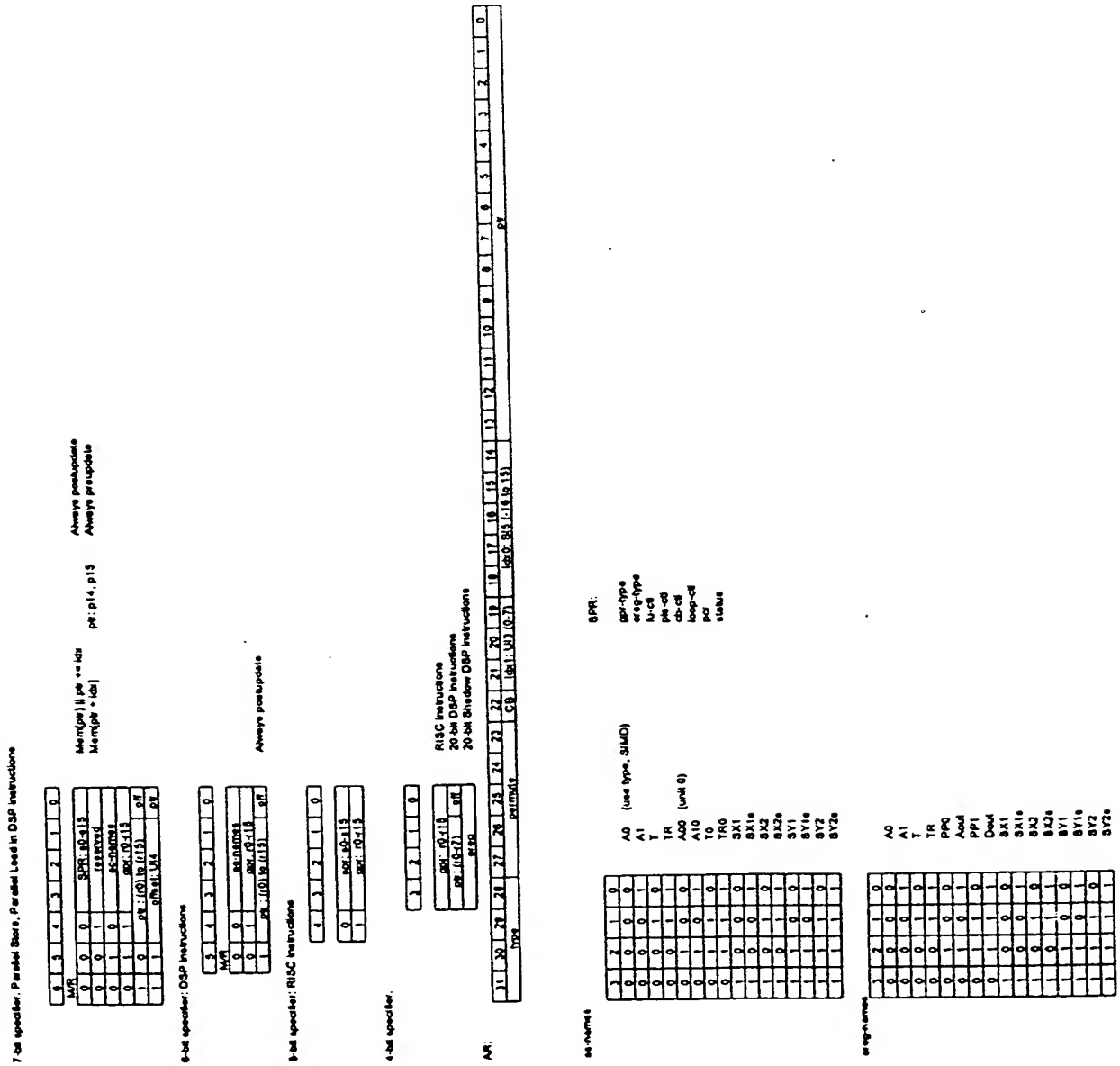

FIG. 6I

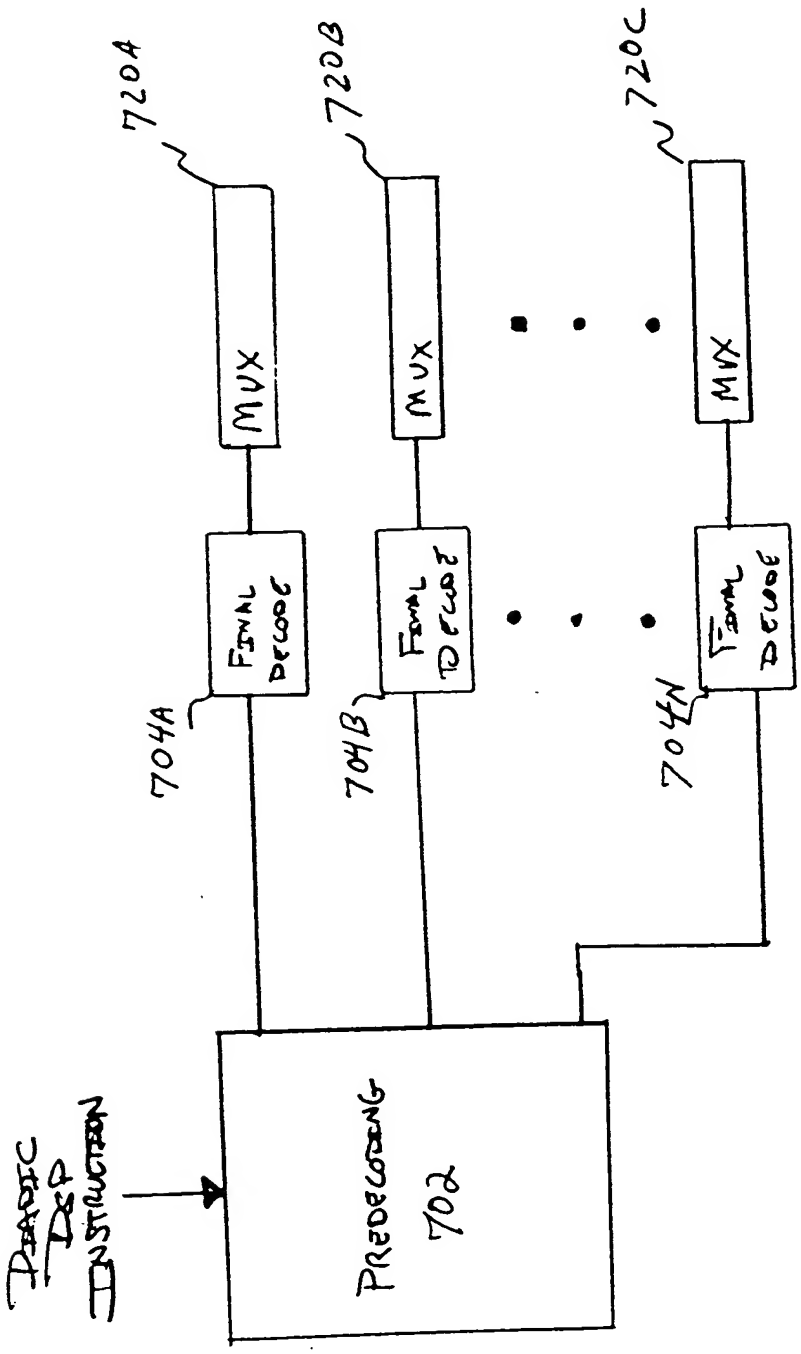
720A
720B
720C
MUX
MUX
MUX
FINAL DECODE
FINAL DECODE
FINAL DECODE
704A
704B
704N
PREDECODING
702
PERIODIC DSP INSTRUCTION

FIG. 7